A FRAMEWORK FOR SURVEILLANCE VIDEO INDEXING AND RETRIEVAL

Thi Lan Le, Alain Boucher, Monique Thonnat, François Bremond

A FRAMEWORK FOR SURVEILLANCE VIDEO INDEXING AND RETRIEVAL

Thi-Lan Le[1,2]*, Alain Boucher*[3]*, Monique Thonnat*[1]*, François Brémond* [1]

[1] PULSAR, INRIA
2004 route des Lucioles, B.P. 93
06902 Sophia Antipolis
France

[2]International Research Center MICA
Hanoi University of Technology
Hanoi
Viet Nam

[3]Equipe MSI
Institut de la Francophonie
pour l'Informatique, Hanoi
Viet Nam

{Lan.Le_Thi, Monique.Thonnat, Francois.Bremond}@sophia.inria.fr
Alain.Boucher@auf.org

ABSTRACT

We propose a framework for surveillance video indexing and retrieval using objects features and semantic events. In this paper, we focus on the following features: (1) combine recognized video contents (at higher level and output from a video analysis module) with visual words (at low level computed over all the raw video frames) to enrich the video indexation in a complimentary way; using this scheme user can make queries about objects of interest even when the video analysis output is not available; (2) support an interactive module that allows users to formulate easily their queries by different ways (existing indexed objects, subimage example, feature generation); more specifically, interactive feature generation (currently color histogram and trajectory) gives a facility for users to make queries at different levels according to the a priori available information and the expected results from retrieval; (3) develop a relevance feedback module adapted to the proposed indexing scheme (recognized video content and visual words) and the specific properties of surveillance videos for the video surveillance context. Results emphasing these three aspects proves a good integration of video analysis for video surveillance and interactive indexing and retrieval.

1. INTRODUCTION

The increasing number of cameras provides a huge amount of video data. Associating to these video data retrieval facilities become very useful for many purposes and many kinds of staff. While some approaches have been proposed for video retrieval in meetings, movies, broadcast news, and sports [1], very few work has been done for surveillance video retrieval [2], [3], [4]. Current achievements on automatic video understanding [5] such as object detection, object tracking and event recognition, though not perfect, are reliable enough to build efficient surveillance video indexing and retrieval systems. To solve the surveillance video indexing and retrieval problem, we need to have both a rich indexing and a flexible retrieval enabling various kinds of user queries.

We have proposed [6] a framework for surveillance video indexing and retrieval. This framework is based on a video analysis engine and a query language. The proposed framework enables users to express their queries by the proposed query language and to retrieve the recognized video contents provided by a video analysis module event with the imprecise and incomplete indexing. In this paper, we extend the existent framework for the surveillance video indexing and retrieval: (1) enrich the indexing by combining the recognized video contents (at higher level and output from a video analysis module) with visual words (at low level computed over all the raw video frames); (2) support an interactive module that allows users to formulate easily their queries by different ways (existing indexed objects, subimage example, feature generation); (3) develop a relevance feedback module adapted to the proposed indexing scheme (recognized video content and visual words) and the specific properties of surveillance videos for the video surveillance context.

The organization of the paper is as follows. Section 2 presents the proposed approach that consists of the indexing phase and the retrieval phase. The indexing phase and the retrieval phase are described in the section 3 and section 4 respectively. The results of the proposed approach with video coming from the CARETAKER project (Content Analysis and REtrieval Technology to Apply Extraction to massive Recording) are presented in the section 5. Finally we present a conclusion and future work in section 6.

2. THE PROPOSED APPROACH

Figure 1 shows the global architecture of the proposed approach. This approach is based on an external **Video Analysis module** and on two internal phases: an **indexing phase** and a **retrieval phase**.

The external Video Analysis module performs tasks such as mobile object detection, mobile object tracking and event recognition. The results of this module are some Recognized Video Contents. These Recognized Video Contents can be

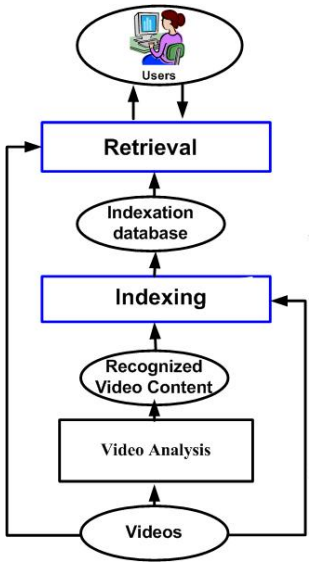

Fig. 1. The global architecture of the proposed approach. This approach is based on an external Video Analysis module and on two internal phases: an indexing phase and a retrieval phase.

physical objects, trajectories, events, scenarios, etc.

For the indexing phase, we already have proposed [6] an indexing approach based on two types of Recognized Video Contents. In this paper, we extend this approach to take into account the missing video contents (i.e. not detected or recognized by the Video Analysis module). We add a generic indexing based on the visual words. Therefore, the new indexing scheme is a combination of the specialized recognized video contents and the generic visual words indexing. Outputs of the indexing phase (Indexation database) become inputs of the retrieval phase.

In the retrieval phase, in order to support a means for retrieving the data, we have proposed a query language called SVSQL (Surveillance Video Structured Query Language)[6]. Users formulate their queries by using the proposed language. They can also feed into the query their example images that they have. In this paper, we enrich the variety of query by allowing users to generate the features (currently color histogram and trajectory). The feature generation is suitable for the scenario in which users do not have example images. However, they have some ideas about the feature. For example, they want to know whether a red car appears in the video but they do not have any example image containing the red car. In the previous work, as the retrieved results are returned to users, no user interaction is allowed. In this work, we attempt to put users in the retrieval loop so that the retrieved results can be improved based on the users' feedback.

3. INDEXING

The indexing phase (Fig. 2) takes either results from the Video Analysis module (the Recognized Video Content) or the raw frames as input data. The indexing phase has two main tasks: **feature extraction** and **data indexing**. It performs feature extraction to complete the input data by computing missing features and data indexing using a data model. According to the input (physical objects or frames), the feature extraction task computes the low level features such as color histogram,... for physical objects or build visual words for frames.

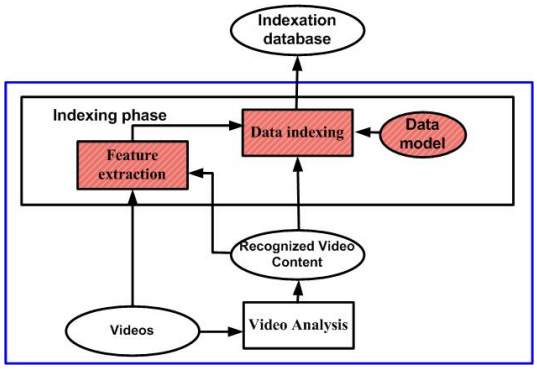

Fig. 2. The indexing phase of the proposed approach. Architecture of the indexing phase is the same as that in [6]. However, the feature extraction, data indexing tasks are extended to work with frames.

3.1. Data model

The data model contains two main types of components: **Recognized Video Content** (**Physical objects**, **Events**) and **Frames**. The physical objects are all the objects of the real world in the scene observed by the camera. One physical object can be a contextual object or a mobile object. We are recently interested in mobile objects. In video surveillance, different kinds of states and events can be defined. In order to facilitate the query making, we group them all into one sole 'Events' concept. The frames are the raw frames extracted from video. Without losing information, we take one frame per second from the video. In the following data model, an attribute written into brackets mean that it is optional, i.e. it may be used according to application needs.

3.1.1. Recognized Video Content

Currently, Physical_objects is defined as follows:

Physical_objects(ID, Class, [Name], 2D_positions, 3D_positions, MBRs, Time_interval, Features)

where ID is the label, Class is the class that the physical object belongs to, 2D_positions and 3D_positions are posi-

tions in 2D and 3D of the physical object, MBRs are the minimum bounding box, Time_interval indicates frames in which the physical object exists. The Features is currently defined as Histograms, Trajectory. Other features are certainly possible to add. We describe briefly these features as follows:

- **Color histogram**: Color histogram[7] is a common feature used for image and video indexing and retrieval based on color information.
- **Object's trajectory**: Two methods of trajectory representation are employed in our approach. One is interested in the starting point and the ending point of the object's trajectory. In [8] Patino et al. shows that the starting point and the ending point can cluster objects' trajectories in several meaningful classes. While the first method is not interested in trajectory's form, the second method do. The second method enable to retrieve trajectories by their forms. Currently, we use the LSCF (Least Square Curve Fitting) and Symbolic representation [9] to analyze the form of trajectory.

The Events are the recognized events in the video database and are defined as follows:

Events(ID, Name, Confidence_value, Involved_Physical_objects, [Sub_events], Time_interval)

where ID is the label of the event, Name is the name of the event, Confidence_value is the confidence degree of event recognition, Involved_Physical_objects is the physical objects involved in the event, Sub_events is the sub events of the event, Time_interval indicated frames in which the event is recognized.

3.1.2. Frames

The Frames are the frames extracted from the Video. This component is a complementary component for Recognized Video Content. It assures that the approach is able to answer users queries about objects of interest even the Video Analysis module is not perfect. Frames is defined as follows:

Frames(ID, Features)

where ID is the label of the frame. The Features is currently defined as a list of visual words, though other features are certainly added. In the context of video surveillance, cameras are fixed however objects are mobile objects. They change therefore their appearances and occlude other objects. The visual words proposed in [10] prove that they are able to retrieve objects successfully despite changes in viewpoint, illumination, and partial occlusion. Moreover, an efficient retrieval is archived by employing the methods from statistical text retrieval on the visual words, including inverted file systems, and text and document weightings.

The visual words are computed exactly as presented in [10]. After computing the visual words, a frame is represented as follows:

$$v_d = (t_1, ..., t_i, ..., t_V)^T. \quad (1)$$

of weighted word frequencies with components:

$$t_i = \frac{n_{id}}{n_d} log \frac{N}{n_i} \quad (2)$$

where V is the number of visual words, n_{id} is the number of occurrences of word i in frame d, n_d is the total number of words in the frame d, n_i is the number of frames containing term i and N is the number of frames in the whole database. The weighting is a product of two terms: the word frequency $\frac{n_{id}}{n_d}$, and the inverted document frequency $logN/n_i$.

4. RETRIEVAL

Figure 3 presents the retrieval phase of the proposed approach. The retrieval phase has 7 tasks: **query formulation**, **query parsing**, **query matching**, **result ranking**, **result displaying**, **feature extraction**, and **SubImage selection** as presented in [6] and two new tasks: **feature generation** and **relevance feedback**.

Users submit queries by using the query language (query formulation task). They can also feed an example image by using SubImage selection task. The feature extraction task aims at computing the same features as the indexing phase. The query parsing tasks analyzes the syntax of query while the query matching task compares the query with the indexed information in the database. The goal of the result ranking and result displaying tasks is to rank results based on their similarity and to display the retrieved results to users.

The **feature generation** and **relevance feedback** are new tasks. The feature generation aims at creating query with the generated features as input. The relevance feedback task allows the system to learn from the user's feedback in order to improve retrieval results. Our preliminary work for relevance feedback based on MIL (Multiple Instance Learning) is presented in this paper.

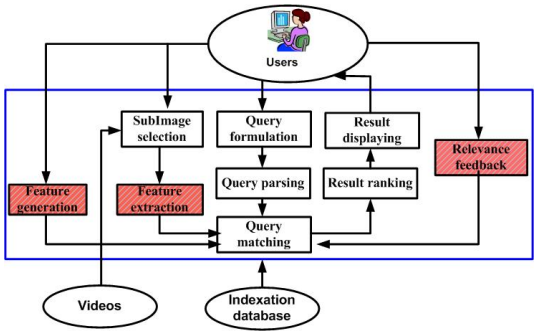

Fig. 3. The retrieval phase of the proposed approach. The feature generation and relevance feedback are new tasks. The feature extraction task is extended in order to compute the visual words for a generic indexing

4.1. Query language

We have proposed [6] a query language for retrieving the surveillance video. The syntax of a query expressed by this query language is the following:

SELECT <Select list > FROM < Database > WHERE <Conditions >

Where: **SELECT**, **FROM**, **WHERE** are keywords for a query and they are mandatory :

- **Select list** specifies the returned results. It may be either whole Physical_objects (or attributes) or whole Events (or attributes) or Frames. We have implemented an aggregate operator COUNT that counts the number of the returned results.
- **Database** specifies which parts of the video database are used to check the <Condition>. It can be either * for the whole video database or a list of named subparts. This is interesting for the surveillance video retrieval because the video database can be divided into several parts according to time or to location. It enables to accelerate the retrieval phase in the case that the users know which parts of the video database they are interested in.
- **Conditions** specifies the conditions that the retrieved results must satisfy. The users express their requirements by defining this component.

We give here some examples of queries expressed in this language. For more detail about this query language, please read our paper [6]. An example expressed by this language returns the Event type of Recognized Video Content: Find Close_to_Gates events occurring in videos from all databases.

*SELECT e FROM * WHERE ((e: Events) AND (e's Name = "Close_to_Gates"))*

where e is a variable of Events, $e's\ Name$ is an access function that gets attribute $Name$ of e.

Another example expressed by this language returns the Physical object type of Recognized Video Content based on relationship between the Event and the Physical object types: Find the Physical_objects in the database named Video_Database that are close to gates.

*SELECT p FROM * WHERE ((e: Events) AND (p: Person) AND (e's Name = "Close_to_Gates") AND (p involved_in e))*

where p is a variable of Person and the involved_in predicate determines whether p involves in e.

4.2. Recognized Video Content retrieval

4.2.1. Recognized Video Content retrieval from Subimage

In the previous section, the queries that return the Recognized Video Contents based on known information of these contents are described. However, if users have an example image containing the object of interest and they are interested in the Recognized Video Contents that are similar to the example image in some way. The SubImage type is supported in the proposed query language. In the following example, users want to retrieve the physical objects in database that named Video_Database that are similar to a given image.

SELECT p FROM Video_Database WHERE ((p: Physical_objects) AND (i: SubImage) AND (i keypoints_matching p))

where p is a variable of Physical_objects, i is a variable that will be set by an image example, $keypoints\_matching$ is a non temporal predicate.

4.2.2. Feature generation

The example images in the previous section are not always available. In that case, users do not have any example image. However, they know some characteristics about the contents they search. Therefore, the feature generation task is developed in our framework to help users to submit their queries. Currently, the feature generation task enables to generate color histogram and trajectory. Because these two characteristics are the most interesting characteristics of objects for users when they retrieve a video database. Moreover, the color histogram and trajectory are already computed in the Features attribute for physical objects in the data model, the generated characteristics are compatible with that in Indexation Database. The matching task does not need to change when working with this type of query.

Color histogram generation: Users can choose one or more colors they want from a color palette. The chosen color is represented by 3 color components (r,g,b). Three Gaussian distributions with $\mu_r = r$, $\mu_g = g$, $\mu_b = b$, $\sigma_r = \sigma_g = \sigma_b$ are generated. Figure 4 illustrates this process. In the experiment, we fix the $\sigma = 20$ parameter. The choice of σ does not change much the retrieval results because for the retrieval, we are interested in the rank of the retrieved results (not exact value of distance). With change of σ, the distance will be changed, but the rank does not change. As persons move in the scene, their colors may be changed. However, these persons are indexed in a number frames and the importance is not to retrieve all of frames but can retrieve a frame in these frames. In this paper, we use the Gaussian distribution for each color component because users can not provide a precis color so the Gaussian distribution allows to take into account the colors neighbor of the chosen color. The advantage of this allows to generate easily color histograms with an interface. Users can choose many colors as they want, the manner to generate the histogram and to do matching between histograms does not need to change. Instead of using machine learning algorithm to learn the color concepts (that can loose the information), with this approach the query is made with semantic concepts of users while the matching is done at the low level. An ex-

ample of query using color histogram generation is given as follows:

SELECT o FROM CARE_2 WHERE ((c: SubImage) AND (o: Physical_objects) AND (c color_similarity o))

User defines a color of interest, the feature generation task create a color histogram based on the algorithm in Fig. 4. The query matching task compare the generated histogram with histograms of physical objects.

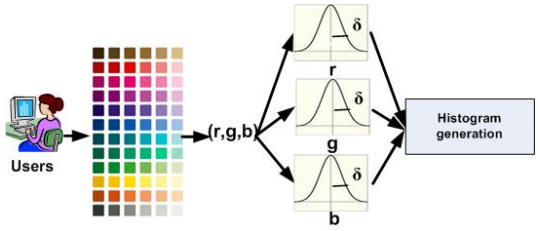

Fig. 4. Color histogram generation process. Users can choose one or more colors they want from a color palette. Three Gaussian distributions are created for the chosen color.

Trajectory generation: Corresponding to two trajectory representations, there are two ways for creating a trajectory (as illustrated in Fig. 5). In the first way (a), users specify two points in an image of scene of video (for each video, we will display the image of scene so that users can define object positions in this image). The generated trajectory is represented by its starting and ending points. The matching is done by computing the Euclidean distance between the generated starting and ending points and those of physical objects in database. In the second way (b), the designed form of trajectory is analyzed by the Least square curve fitting and the Symbolic representation. The results of this analysis are matched with trajectories in database. A query using the generated trajectory is described as follows:

SELECT o FROM CARE_2 WHERE ((o: Physical_objects) AND (i: SubImage) AND (o trajectory_similarity_start_end p))

For this query, user specifies a query trajectory by its starting point and ending point. Retrieved results of this query are physical objects whose trajectories are similar to the query trajectory. The trajectory_similarity_start_end predicate determines the similarity between two trajectories based on the starting and ending points.

4.3. Retrieval of non recognized objects

When working with the Recognized Video Contents, we suppose that they are already recognized by the Video Analysis Module. However, the Video Analysis Module is not perfect. In order to allow users to retrieve their objects of interest, a generic indexing is computed over video frames. In the surveillance video context, the objects (persons) are moving objects. They change their appearance in the time. The generic

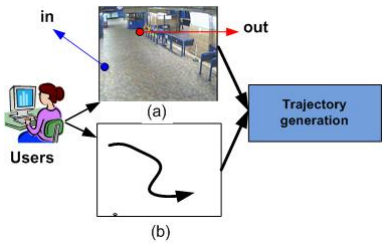

Fig. 5. There are two ways to create a trajectory (a) trajectory is represented by its starting point and ending point. Users specify two points in an image of scene of video (b) trajectory is represented by its form. Users designs a trajectory form.

indexing must be able to find objects of interest in frames with an instance of their appearance (in an example image). In order to cope with this problem, we extend the indexing and retrieval capacity by using the visual word technique in [10]. The process presented in the section 3.1.2 is applied on the example SubImage. This SubImage is represented as a vector v_q (Eq. 1). The appear_in predicate is determined according to the distance between two vectors v_d and v_q. This distance is defined as follows:

$$f_d = \frac{v_q^T v_d}{\sqrt{v_q^T v_q}\sqrt{v_d^T v_d}} \quad (3)$$

The process to determine appear_in predicate is illustrated in Fig. 6. Note that visual detection in this figure belongs to the feature extraction task. The query presented as follows retrieves frames containing object of interest that is specified in an example SubImage.

SELECT f FROM CARE_2 WHERE ((f: Frames) AND (i: SubImage) AND (i appear_in f))

4.4. Relevance feedback on the visual words

The combining of indexing based on the recognized video contents with a generic indexing based on the visual word technique allows to answer request of users in various conditions (the Video Analysis module is perfect or not perfect or not available). This combining enriches the indexing by the bottom-up approach. A top-down approach that takes user feedback in order to improve the retrieval results (short term) and to complete the indexation database (long term) must be considered. Currently, we concentrate on the relevance feedback for the generic indexing because this indexing retrieve object of interest in the case that the Video Analysis module is not perfect or not available to detect objects. It can cause irrelevant results and is necessary to do the relevance feedback.

As previously presented, the generic indexing is based on the visual word matching. The result (is frame) is judged positive or negative result. However, there are not information

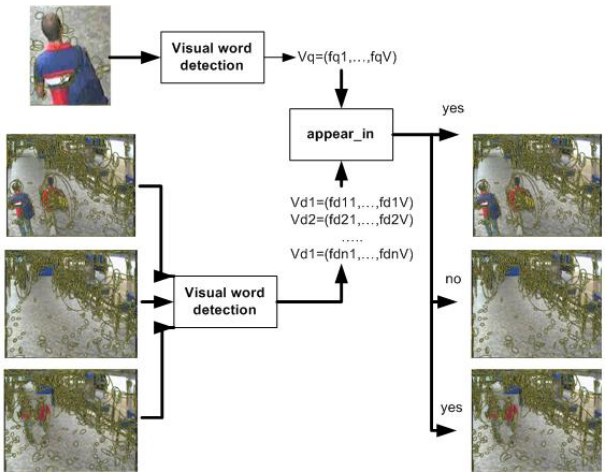

Fig. 6. The process for determining the appear_in predicate. The visual detection box belongs to the feature extraction task. The object of the query was not previously detected by the video analysis module but can still be retrieved by the retrieval module.

about visual words (users are not able to judge whether a visual words is positive or negative). Therefore, each returned frame can be considered as a bag of visual words and the relevance feedback on these frames become MIL problem. This problem becomes Multi-Instance Learning problem that is defined below:

Definition: Given a set of training examples $T < B, L >$ *where* $B = B_i(i = 1, ..., n)$ *is a set of n bags and* $L = L_i(i = 1, ..., n)$ *is a set of labels of the corresponding bags.* $L_i \in 1(Positive), 0(Negative)$. *The goal of MIL is to identify the label of a given instance in a given bag.*

In the scenario of MIL, the labels of individual instances are not available, instead the bags are labeled. If the bag label is positive, there exists at least one positive instance in that bag. If the bag label if negative, all instances in that bag are negative. In this paper, we use the MILL toolkit [1] to do the relevance feedback.

5. EXPERIMENTAL RESULTS

5.1. Databases

We use the 2 hours-video from the CARETAKER[2] (Content Analysis and REtrieval Technology to Apply Extraction to massive Recording) project. The Video Analysis module (object detection and object tracking) developed by ORION team is applied on this video. The 71 objects are detected from this video. Note that the event recognition is not applied in this video and the results of Video Analysis module are used only for color histogram and trajectory evaluation. In the frame retrieval and relevance feedback, we employ the raw frames of video (corresponding the case that the Video Analysis module is not available).

Table 1. Four experiments: color histogram, trajectory, frame retrieval and relevance feedback with their test queries and the size of database.

Name of test	Nbqueries	Size's database	Results
Color histogram	7	5597(frames)	Fig. 7
Trajectory	7	71 (trajectories)	Fig. 9
Frame retrieval Relevance feedback	8	298 (frames)	Fig. 11

5.2. Evaluation

5.2.1. Color histogram evaluation

In order to evaluate the queries based on color information, we create 7 queries for 7 colors Black, White, Red, Blue, Green, Yellow, Violet. In the ground truth, among blobs of the Physical_objects, we decide the blob whose color is similar to the color specified in the query. The performance is evaluated by the precision/recall graph. The obtained precision/recall graphs for these 7 queries are shown in Fig. 7. This obtained result shows that this approach supports for users a possibility to submit a query based on color information with a good result even they do not have any example at their hands.

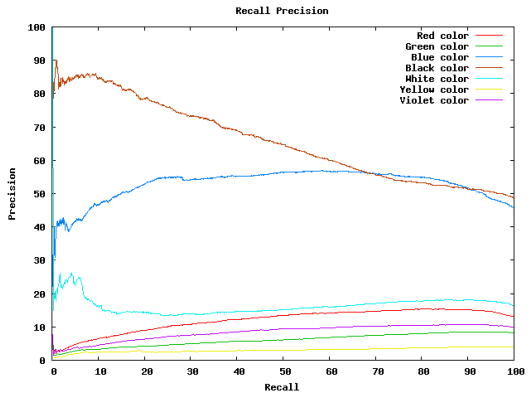

Fig. 7. Recall/precision for query with 7 main colors: Black, White, Red, Blue, Green, Yellow, Violet.

[1] Jun Yang, MILL: A Multiple Instance Learning Library, http://www.cs.cmu.edu/ juny/MILL.

[2] http://www.ist-caretaker.org/

5.2.2. Trajectory evaluation

Users can submit a query by defining a starting point and an ending point. In order to evaluate the proposed approach, we generate several trajectory queries by choosing the starting and ending points from the scene's image. These queries are shown in Fig. 8. We make manually the ground truth in which we specify which trajectories in the database are similar to the query (basing on the starting point and the ending point) Fig. 9 presents the obtained results of the trajectory retrieval based on the starting point and the ending point. The good results show that users can retrieve successfully objects based on their changes of zone.

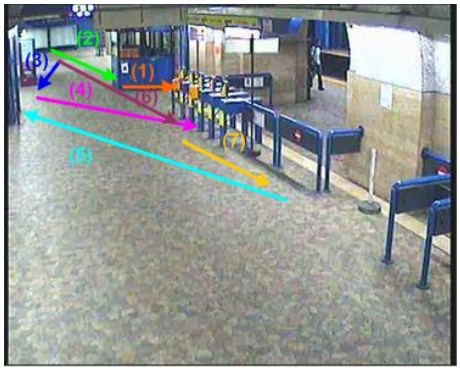

Fig. 8. Some trajectory queries based on the starting point and ending point. Users define these points on scene's image.

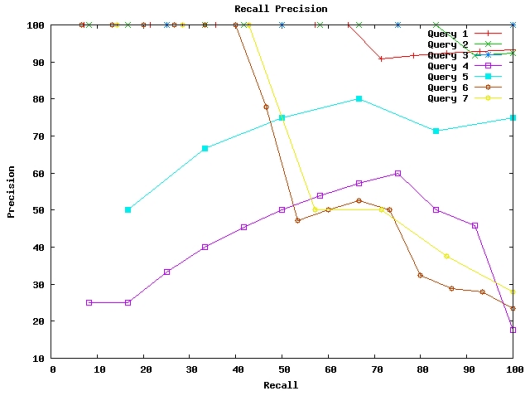

Fig. 9. The obtained recall and precision graphs for the queries in Fig. 8.

5.2.3. Non recognized video content retrieval and relevance feedback evaluation

In this section, we evaluate non recognized video content retrieval performance over the entire video. The object of interest is specified by the user as a sub part of frame (is represented by SubImage type). We submit 8 times the query. At each time, we decide a different object of interest as presented in Fig. 10. The pairs of queries (1,3), (4,7) and (5,8) indicate the same person with different appearances (front's appearance and back's appearance). A visual dictionary is

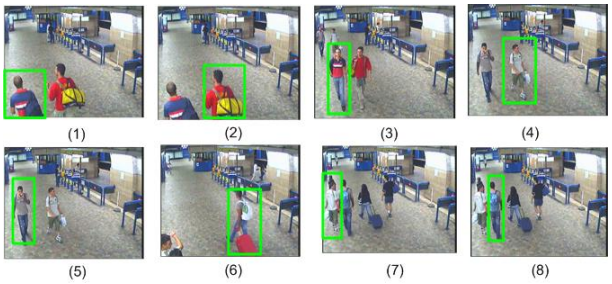

Fig. 10. Query frames with outlined query regions for the eight test queries.

built from the detected affine covariant regions on the frames. The number of visual words is 500. We apply the stop list algorithm like [10] in order to remove the most frequent visual words that occur in almost all frames. The top 5% and bottom 5% are removed. These words correspond to the objects in the empty scene. The acceptable results are obtained (Fig.

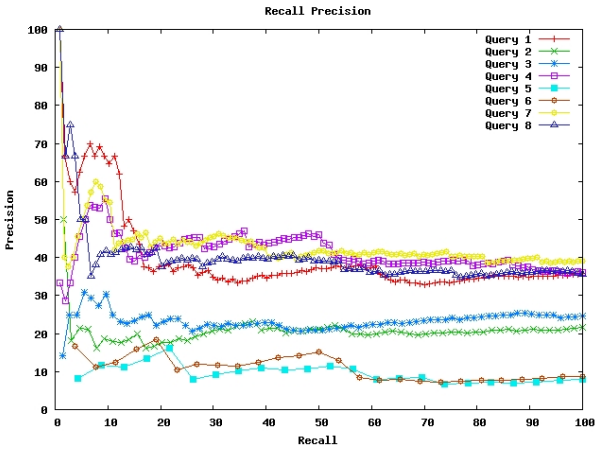

Fig. 11. The obtained recall/precision graphs for the 8 test queries in Fig. 10.

11) even though objects change noticeably their appearance in the scene. Moreover, users are often interested in the first results that are successful retrieved (high value of precision when the corresponding recall is from 0 to 20). Figure 12 presents the obtained results of relevance feedback after one loop of relevance feedback. Frames are judged as relevant and irrelevant results. The results are improved after one loop of relevance feedback.

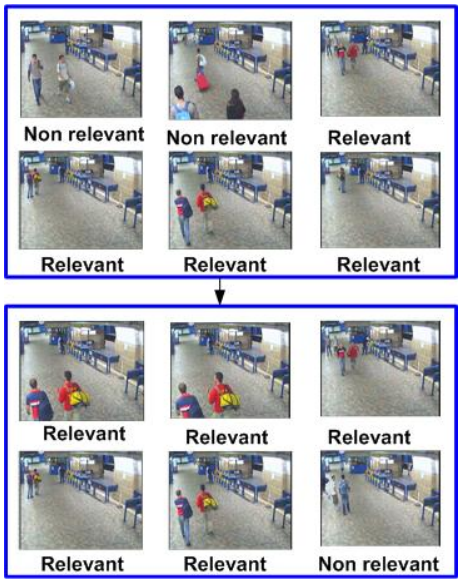

Fig. 12. The obtained results after one loop of relevance feedback for query 1 in Fig. 10.

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented a mixed framework from the pure retrieval approach and the pure recognition approach. According to the types of queries and desirable results, the pure retrieval approach (list of ranked results) or the pure recognition approach (list of non ranked, fixed results) or a mixed approach is used (list of ranked, fixed results). This allows to take advantage of retrieval approach to deal with errors of recognition technique. A combining indexing that employs both the recognized video contents and frames (with visual words) gives a good retrieval performance. Because, the recognized video content provides a high level and richer semantic information about the videos. Indexation by visual words is done at low level but provides a complementary way to retrieve missed objects by the video analysis module. We have presented in section 5 separate results of two indexing approach aspects (one is based on the recognized video content and the other is based on the visual words). Results of the combination of two aspects must be provided. Moreover, a preliminary work for the relevance feedback is presented. A complete relevance feedback module and experimental results will be available in the future.

7. REFERENCES

[1] Ziyou Xiong, Xiang Sean Zhou, Qi Tian, Yong Rui, and Huang TS, "Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 18–27, 2006.

[2] A. Hampapur, L. Brown, J. Connell, A. Ekin, N. Hass, M. Lu, H. Merkl, S. Pankanti, A. Senior, C-F. Shu, and Y-L. Tian, "Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking," *IEEE Signal Processing Magazine*, vol. 50, no. 22, pp. 38–51, 2005.

[3] W. Hu, D. Xie, Z. Fu, W. Zeng, and Maybank S., "Semantic-based surveillance video retrieval," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 1168–1181, 2007.

[4] S. Calderara, R. Cucchiara, and Prati A., "Multimedia surveillance: Content-based retrieval with multicamera people tracking," in *VSSN06*, Santa Barbara, California, USA, 27 October 2006.

[5] Paolo Remagnino, Graeme A. Jones, Nikos Paragios, and Carlo S. Regazzoni, *Video Based Surveillance Systems Computer Vision and Distributed Processing*, Kluwer Academic Publishers, 2002.

[6] Thi-Lan Le, Monique Thonnat, Alain Boucher, and François Brémond, "A query language combining object features and semantic events," in *The 14th International MultiMedia Modeling Conference (MMM)*, Kyoto, January 2008.

[7] M.J. Swain and D.H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[8] J.L Patino, H. Benhadda, E. Corvee, F. Bremond, and M. Thonnat, "Video-data modelling and discovery," in *International Conference on Visual Information Engineering VIE 2007*, London, UK, 25th -27th July 2007.

[9] Thi-Lan Le, Alain Boucher, and Monique Thonnat, "Subtrajectory-based video indexing and retrieval," in *The 13th International MultiMedia Modeling Conference (MMM)*, Singapore, January 2007, pp. 418–427.

[10] J. Sivic and A. Zisserman, "Video google: Efficient visual search of videos," *Toward Category-Level Object Recognition*, vol. LNCS 4170, pp. 127–144, 2006.